

IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. **(currently amended)**: A graphics processor, comprising:
a plurality of parallellized graphics computational units; and
one or more task allocation units programmed to bypass defective ones of said graphics computational units within said[[groups]] plurality, and to distribute incoming tasks only among operative ones of said graphics computational units.
2. (canceled)
3. (previously presented): The graphics processor of claim 1, wherein each of said parallellized graphics computational units also includes respective multiple vertex processors.
4. (previously presented): The graphics processor of claim 1, wherein each of said parallellized graphics computational units also includes respective texturing pipelines.
5. (previously presented): The graphics processor of claim 1, wherein each of said parallellized graphics computational units also includes a respective memory controller.
6. (canceled)

Amendment – Serial No. 10/086,980.....Page 2

7. (original): A method of 3D graphics rendering which comprises: using a task allocation unit and parallellized graphics computational units with relations as recited in claim 1.
8. (previously presented): The graphics processor of claim 1, wherein one or more of said parallellized graphics computational units operate with no more than 4 operative vertex processors.
9. (previously presented): The graphics processor of claim 1, wherein one or more of said parallellized graphics computational units operate with no more than 4 operative texturing pipelines.
10. (previously presented): The graphics processor of claim 4, wherein said texturing pipelines also include a shading unit and a texture filter unit.
11. (previously presented): The graphics processor of claim 4, wherein said texturing pipelines also include a shading unit and a primary texture cache.
12. (previously presented): A method of 3D graphics rendering, comprising the actions of:
 - providing a plurality of parallellized graphics computational units;
 - bypassing defective ones of said units, and
 - distributing incoming tasks only among operative ones of said units.
13. (previously presented): The method of claim 12, wherein each of said parallellized graphics computational units also includes respective multiple vertex processors.

14. (previously presented): The method of claim 12, wherein one or more of said parallellized graphics computational units operate with no more than 4 operative vertex processors.
15. (previously presented): The method of claim 12, wherein each of said parallellized graphics computational units also includes respective multiple texturing pipelines.
16. (previously presented): The method of claim 12, wherein one or more of said parallellized graphics computational units operate with no more than 4 operative texturing pipelines.
17. (previously presented): The method of claim 12, wherein each of said parallellized graphics computational units also includes a respective memory controller.
18. (previously presented): The method of claim 15, wherein said texturing pipelines also include a shading unit and a texture filter unit.
19. (previously presented): The method of claim 15, wherein said texturing pipelines also include a shading unit and a primary texture cache.

20. (previously presented): A computer graphics system comprising:
means for providing a plurality of parallellized graphics computational units;
means for bypassing defective ones of said units, and
means for distributing incoming tasks only among operative ones of said units.
21. (previously presented): The system of claim 20, wherein each of said parallellized graphics computational units also includes respective multiple vertex processors.
22. (previously presented): The system of claim 20, wherein one or more of said parallellized graphics computational units operate with no more than 4 operative vertex processors.
23. (previously presented): The system of claim 20, wherein each of said parallellized graphics computational units also includes respective multiple texturing pipelines.
24. (previously presented): The system of claim 20, wherein one or more of said parallellized graphics computational units operate with no more than 4 operative texturing pipelines.
25. (previously presented): The system of claim 20, wherein each of said parallellized graphics computational units also includes a respective memory controller.

26. (previously presented): The system of claim 23, wherein said texturing pipelines also include a shading unit and a texture filter unit.
27. (previously presented): The system of claim 23, wherein said texturing pipelines also include a shading unit and a primary texture cache.
28. (previously presented): A method for computer graphics system operation, comprising the actions of:
- providing a plurality of parallellized rendering units;
 - bypassing defective ones of said units, and
 - distributing incoming tasks only among operative ones of said units.
29. (previously presented): The method of claim 28, wherein each of said parallellized rendering units also includes respective multiple vertex processors.
30. (previously presented): The method of claim 28, wherein one or more of said parallellized rendering units operate with no more than 4 operative vertex processors.
31. (previously presented): The method of claim 28, wherein each of said parallellized rendering units also includes respective multiple texturing pipelines.
32. (previously presented): The method of claim 28, wherein one or more of said parallellized rendering units operate with no more than 4 operative texturing pipelines.

33. (previously presented): The method of claim 28, wherein each of said parallelized rendering units also includes a respective memory controller.
34. (previously presented): The method of claim 31, wherein said texturing pipelines also include a shading unit and a texture filter unit.
35. (previously presented): The method of claim 31, wherein said texturing pipelines also include a shading unit and a primary texture cache.
36. (new): A method for improving the yield enhancement of graphic chips, comprising the actions of:
- manufacturing a graphics chip having a plurality of parallelized computational units;
 - testing said graphics chip to determine any non-operative computational units among said plurality of parallelized computational units;
 - recording the results of said testing; and
 - using said chip in a lower performance product if said chip is determined to contain any non-operative computational units.
37. (new): The method of claim 36, wherein a table allocation unit bypasses said non-operative computational units.
38. (new): The method of claim 36, wherein said chip has some but not all pipelines functional.
39. (new): The method of claim 36, wherein said results are recorded on an off-chip configuration memory.

40. (new): A graphics chips, comprising:
- a plurality of parallelized graphics computational units, and
 - a record indicating any non-operative computational units among said plurality of parallelized computational units;
 - wherein said record is used to specify said chip under a secondary part specification if said chip is determined to contain any non-operative computational units.
41. (new): The chip of claim 36, wherein a table allocation unit bypasses said non-operative computational units.
42. (new): The chip of claim 36, wherein said chip has some but not all pipelines functional.
43. (new): The chip of claim 36, wherein said record is on an off-chip configuration memory.
44. (new): A graphics system, comprising:
- a graphics chip comprising plurality of parallelized graphics computational units,
 - a record indicating any non-operative computational units among said plurality of parallelized computational units; and
 - a software device drive which recognizes which of the parallelized graphics computational units are functional and uses only the functional parallelized graphics computational units.